

## PATENT ABSTRACTS OF JAPAN

O A ①  
Ref ③

(11)Publication number : 01-237726

(43)Date of publication of application : 22.09.1989

(51)Int.Cl.

G06F 9/06

G06F 9/44

(21)Application number : 63-063295

(71)Applicant : HITACHI LTD

(22)Date of filing : 18.03.1988

(72)Inventor : ITOU KISHIO

## (54) AUTOMATIC GENERATING SYSTEM FOR SOFTWARE SPECIFICATION

(57)Abstract:

PURPOSE: To allow the maintenance work of a software to be efficient by analyzing plural software products, preparing the specifying information of a high-order specification and converting these specifying information to the formation of the high-order specification.

CONSTITUTION: A JCL (job control language) analyzing part 2 analyzes a JCL from a JCL library 1 and outputs information concerning an input/output file to a JCL analyzing information file 3. Next, a file relation analyzing part 4 inputs the file 3 and outputs the names of a sharing file and a sharing job to a file relating information file 5. Then, a network information preparing part 6 prepares a network, which expresses connecting relation between a job file and a slip, from the information 3 and file 5 and outputs the network to a network information file 7. Finally, an expressing form converting part 8 inputs the file 7 and converts the file to a data flow specification, which is a purpose. Namely, network information is inputted and an arranging position in the specification, which is prepared concerning respective nodes and branches, is determined.



⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A) 平1-237726

⑬ Int. Cl.<sup>4</sup>

識別記号

庁内整理番号

⑭ 公開 平成1年(1989)9月22日

G 06 F  
9/06  
9/44

3 2 0  
3 2 0

B-7361-5B  
E-8724-5B

審査請求 未請求 請求項の数 7 (全8頁)

⑮ 発明の名称 ソフトウェア仕様書の自動生成システム

⑯ 特 願 昭63-63295

⑰ 出 願 昭63(1988)3月18日

⑱ 発 明 者 伊 藤 岸 男 東京都品川区南大井6丁目23番15号 株式会社日立製作所  
大森ソフトウェア工場内

⑲ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑳ 代 理 人 弁理士 小川 勝男 外1名

明 細 書

1. 発明の名称

ソフトウェア仕様書の自動生成システム

2. 特許請求の範囲

1. ソフトウェア仕様を作成する情報処理システムにおいて、ソフトウェア開発において上位仕様書を詳細化することにより得られた少なくとも一つのソフトウェア生産物(プログラム、ジョブ制御文、下位仕様書等)の記述内容を読み取り上位仕様書から引き継がれている内容を出し、抽出内容の内同一項目の記述内容をそれぞれ一項目に集約し、一つの項目に集約した項目と単独で存在する項目をまとめネットワーク状の項目情報を作成し、ネットワーク状に配置された各項目の表現形式を変換し目的とする仕様書を作成することを特徴とするソフトウェア仕様書の生成システム。

2. ソフトウェア生産物より上位仕様書から引き継がれている内容の抽出において、利用者が上位仕様書の情報を導く手段を有することを特徴

とする特許請求の範囲第1項もしくは第2項記載のソフトウェア仕様書の自動生成システム。

3. ソフトウェア仕様書を生成するために充分な情報が得られなかった場合、複数複製のソフトウェア生産物からの情報を抽出する手段を有することを特徴とする特許請求の範囲第1項、第2項のいずれかの項記載のソフトウェア仕様書の自動生成システム。

4. ソフトウェア生産物がジョブ制御文(以下JCLと呼ぶ)であり、上位仕様書としてデータフローダイアグラムを記述したデータフロー仕様書を作成することを特徴とする特許請求の範囲第1項、第2項、第3項のいずれかの項記載のソフトウェア仕様書の自動生成システム。

5. ソフトウェア生産物がジョブ内のプログラム実行順序及びプログラムの入出力ファイルを模式的に記述したシステムフロー仕様書であり、上位仕様書としてデータフローダイアグラムを記述したデータフロー仕様書を作成することを特徴とする特許請求の範囲第1項、第2項、第

3項のいずれかの項記載のソフトウェア仕様の自動生成システム。

6. ソフトウェア生産物がデータベース記述言語によって記述されたデータベース定義情報であり、上位仕様書としてデータベース内の構成要素の関連を表すE. R (Entity, Relation) 仕様書を生成することを特徴とする特許請求の範囲第1項、第2項、第3項のいずれかの項記載のソフトウェア仕様の自動生成システム。

7. ソフトウェア生産物がオンラインプログラムのソースプログラムであり、上位仕様書として画面名称と画面遷移状態を表す画面遷移仕様書を生成することを特徴とする特許請求の範囲第1項、第2項、第3項のいずれかの項記載のソフトウェア仕様の自動生成システム。

### 3. 発明の詳細な説明

(産業上の利用分野)

本発明は、ソフトウェア開発の計算機による支援に係り、特にソフトウェアの保守に好適なソフトウェア仕様書の作成システムに関する。

アの保守を行うには次の問題点が残されている。初期の段階で作成された上位仕様書と、これを詳細化して作成されたソフトウェア生産物(プログラム、ジョブ制御文、下位仕様書)と内容が一致しない場合が生じる。この場合、仕様書は実際のプログラムやジョブ制御文(以下「C1」と呼ぶ)の内容を渡わなくなり、保守が困難になる。

本発明の目的は、仕様書とプログラム、あるいは上位仕様書とこれを詳細化した仕様書の不一致を無くすことにより、ソフトウェアの保守作業を容易にすることにある。

[課題を解決するための手段]

上記目的は、複数の詳細化された仕様書あるいはプログラムやC1などのソフトウェア生産物を解析してその中の上位仕様書を含むべき仕様情報と、抽出情報の内複数の仕様書の内で共通に含まれる仕様情報を統合し、統合されない抽出情報を含めて上位仕様書の仕様情報を作成し、この仕様情報を上位仕様書の形式に変換することにより達成される。

(従来の技術)

一般にソフトウェアの開発では、開発すべきソフトウェアの具備すべき機能を定め、その機能を実現するソフトウェアを段階的に詳細化して設計を進め、最終的にプログラムを作成する方法が用いられる。

設計結果は、詳細化の各段階ごとに設計仕様書として記述され、次の設計段階では、前の段階で作成された仕様書の各要素を詳細化した仕様書が作成される場合が一般的である。

これらの仕様書やプログラムの作成を効率化するために各種の仕様書のエディタや、プログラムエディタなどの開発支援ツールが開発されている。これらの方法については例えば「日経コンピュータ」1986年7月7日号P121～P136「システムフローを自動生成するソフトBAGLE2」に見られる通りである。

[発明が解決しようとする課題]

従来、新規にソフトウェアを開発する場合には有効であるが、これを用いて開発したソフトウェ

もし、一種類のソフトウェア生産物内の仕様情報のみでは上位仕様書を作成するために必要な情報が得られない場合、他の下位ソフトウェア生産物の情報を解析しその中の情報を前記抽出情報に付加する手段を付加することにより達成できる。

また、ソフトウェア生産物内の情報のみでは上位仕様書を作成できない場合、下位の仕様情報に対し利用者が上位仕様書を作成するために必要な情報を加えることにより達成できる。

[作 用]

上記手段により、下位の仕様書やプログラム等のソフトウェア生産物より、その内容と一致する上位仕様書を効率よく作成することができる。

これにより、仕様書間あるいは仕様書とプログラム間の不整合をなくすことができるため、ソフトウェア保守作業におけるソフトウェアの理解を効率よく行うことができる。

(実施例)

以下、本発明の実施例を図を用いて説明する。本実施例は、計算機上での処理単位となるジョ

プ内での(1)プログラムの実行順序及び(2)プログラムに対する入出力装置を記述するジョブ制御文(以下JCLと呼ぶ)からジョブ間のデータの流れを表す仕様書であるデータフロー仕様書を生成する方法について述べる。データフロー仕様書はJCLの上位仕様書であり、公知のデータフローダイアグラムで表視する。

第1図は、本発明の実施例を実現する機能ブロック図を示す図である。第2図は、第1図の実施例を実現する装置のハードウェア装置の構成図を示し、CPU101、メモリ102、外部記憶装置103、入力装置104および表示装置105からなる。

第1図のJCLライブラリ1は、データフロー仕様書の生成に使用するJCL群を納めたファイルを示す。JCL解析部2はJCLライブラリ1よりJCL群を入力し、入力した各JCLについて下掲規則によりJCLを解析し、その中の当該ジョブの入出力ファイルに関する情報をJCL解析情報ファイル3へ出力する。

されているファイルのファイル名称は17に示すFILEオペランドにより"FILE1"と定義されている。そして18に示すI/Oオペランドにて当該ファイルが、本プログラムに対し入力ファイルであるか出力ファイルであるかが示されている。本例の場合このI/Oオペランドにて入力ファイルであることを示す"INPUT"が記述されている。ASN文12も同様に入出力装置として磁気ディスク装置が割り当てられており、そのファイル名称が"FILE2"であるが、ファイルの入出力規則を記述するI/Oオペランド19に出力ファイルであることを示す"OUTPUT"が記述されている。ASN文13では、入出力装置の識別を記述するEQUIPオペランド20にてプリンター装置を表す"PRINTER"という記述がされており、装置名称21に示すOUTCLASSオペランドにて"A"と定義されている。文14は、当該ジョブにて実行するプログラムの名称を記述する文であり、本例では、"PROGRAM1"というプログラムが実行される。文15

JCL解析規則は、次の通りである。

- (1) 入出力装置のハードウェア種別としてプリンターが割り当てられている場合には、標準出力として解析情報を出力する。
- (2) 入出力装置のハードウェア種別としてプリンター以外が割り当てられている場合には、ファイル名称と入出力区分を解析情報として出力する。

第3図はJCLライブラリ1に蓄積されているJCLの例であり、第4図は第3図のJCLを解析してJCL情報解析ファイル3に蓄積されるJCL解析情報例である。

第3図に示すJCL1では、10で示すJOB文にて"JOB1"というジョブ名称が与えられている。11から13に示すASN文により当該ジョブが使用する入出力ファイルの記述がなされている。ASN文11では、入出力装置の識別を記述するEQUIPオペランド16にて、磁気ディスク装置を表す"DISK"という記述がされており、この磁気ディスク装置上のファイルに作成

は、ジョブの終端を示す。

同様の解析を第3図のJCL2について施し、出力したJCL解析情報ファイル例が第4図である。

次に、第1図のファイル関連解析部4は、JCL解析情報ファイル3を入力し、複数のジョブ間で同一ファイル名称を利用しているものを抽出し、共用ファイル名称及び、共用を行なうジョブのジョブ名称をファイル関連情報ファイル5へ出力する。

第5図は第4図のJCL解析情報ファイル5を入力した場合の共用ファイル情報の抽出例である。第5図のFILE1 22については、第4図のJCL解析情報ファイルでは、JOB1の入力ファイルのみに利用されているため、当該ファイルの参照ジョブ名称として"JOB1"が登録される。第4図のFILE2 23については、第4図よりジョブ名"JOB1"で更新され、ジョブ名"JOB2"で参照されるため、更新ジョブとして"JOB1"、参照ジョブとして"JOB2"が登録さ

れている。同様に、第5図のFILE3 24は、“JOB2”の出力ファイルとしてのみ利用されているため、更新ジョブ名称として“JOB2”が登録されている。循環出力25は、ジョブ名“JOB1”より出力されていることが第4図のJCL解析情報より判明するため更新ジョブ名称として“JOB1”が登録されている。

次に第1図のネットワーク情報作成部6は、JCL解析情報3とファイル関連情報ファイル5よりジョブ、ファイル及び循環の接続関係を渡すネットワーク情報を作成し、ネットワーク情報ファイル7へ出力する。ここで作成するネットワーク情報は、JCL解析情報中に存在する全てのジョブ、ファイル及び循環をノードとし、ファイルとジョブまたは、ジョブと循環間の参照関係に依存するブランチを持つものとする。また、ファイル関連情報解析部より判明された複数ジョブ間で共用されているファイルに対しては、当該ファイルに属するノードが一つだけ存在するネットワーク情報を作成する。第4図に示すJCL解析情報

と第5図に示すファイル関連情報により作成されるネットワーク情報を図式的に表現した例を第6図に示す。第6図中の丸付文字列、棒付文字列及び二重棒付文字列は、各々ジョブ、ファイル、循環のノードとその名称を意味する。また、矢印は、ジョブとファイルまたは、ジョブと循環間のブランチを示し、矢印の向きが参照方向を表す。第8図のJOB1 26に着目した場合、第4図のJCL解析情報より、入力ファイルとして“FILE1”、出力ファイル及び循環として“FILE2”及び“A”が与えられているため、ジョブを根ノード“JOB1”への入力となるノードとして“FILE1”、ノード“JOB1”からの出力となるノードとしてファイル“FILE2”及び循環“A”が接続されている。“FILE2”27に着目した場合、第4図によればジョブ“JOB1”と“JOB2”で共用しているファイルであるが、ネットワーク情報上は、当該ファイルを示すノードが一つだけ存在していることがわかる。

最後に第1図の表現形式変換部8にて、ネット

ワーク情報ファイル7を入力し、目的とするデータフロー仕様書に変換する。表現形式変換部8では、ネットワーク情報を入力し、各ノード及びブランチについて作成すべき仕様書内の配置位置を決定する。配置位置決定アルゴリズムは公知であり、例えば情報処理1986NO6“応用指向メモリ”に示されている。配置位置決定後、各ノード及びブランチについて表現形式を変換しデータフロー仕様書ファイル8へ出力する。

第7図にジョブとジョブ間のデータの流れをネットワーク形式で表現するデータフローダイアグラムの一種であるDFD(Structured Dataflow Diagram)にて表現データフロー仕様書の例を示す。

以上の実施例では、データフロー仕様書を作成するために必要なファイルの入出力に関する情報が全てJCLより得られる場合であるが、本実施例を適用するコンピュータシステムによっては、JCLのみから十分な情報が得られない場合がある。このような場合に好適な実施例を次に示す。

第8図は本実施例の機能ブロック図を示す。第

8図は第1図とほぼ同様であるが、JCL解析情報ファイル編集部28を加えた点が異なる。

第9図は第8図のJCLライブラリ1に選択されるJCLの例を示す。第9図のJCLは第8図に示す前実施例とほぼ同様であるが、第9図中の“JOB1”実行用JCLの内、下線29で示す“FILE2”に関する入出力区分の記述がなされていない点が異なる。第9図のJCLを第8図のJCL解析部2で解析した場合のJCL解析情報ファイル3の例を第10図に示す。第10図では、ジョブ名称“JOB1”、ファイル名称“FILE2”に関する入出力区分下線部30が空欄となっている。このJCL解析情報だけでは第8図に示すファイル関連情報部4に入力した場合、当該ジョブがファイル参照するかどうかは、更新するかが不明であるためネットワーク情報を構成することができない。そこで、JCL解析情報編集部28にて、入出力区分を手入力することによりネットワーク情報を完成させることができる。ジョブ名称、ファイル名称、循環名称についての

情報も同様にして追加、削除、更新を可能とする。

以下、第1図の実施例と同様の手段で目的とするデータフロー仕様書を作成する。

上記実施例では、データフロー仕様書生成のため不足する情報をJCL解析情報を入手により補填することにより補ったが、JCLライブラリー中に記述されているJCLによって実行される各プログラムの詳細仕様であるプログラムの情報を解析しこの情報により当該不足情報を補う場合の実施例を第11図に示す。

第11図の実施例は第1図の実施例と同様であるが、ソースプログラムライブラリ31、ソースプログラム解析情報32、ソースプログラム解析情報ファイル33及びソースプログラム解析情報統合部34を付加した点異なる。

第11図の34において当該ジョブの使用ファイルの入出力区分の情報を補う。

第12図は、第9図のJCLによって実行されるプログラム“PROGRAM1”のC言語によるコーディング例を示す。第12図では、open

文35から37により当該プログラムの入出力ファイルの割り当てを行なった後データ処理を行っている。ここでは、文35により入力ファイルとして“FILE1”、文36により出力ファイルとして“FILE2”、文37により復読出力ファイルとして“A”を割り当てている。本プログラムは、ソースプログラムライブラリ31に記憶される。

第11図のソースプログラム解析部32では、ソースプログラムライブラリ31より本プログラムを入力し文35から37で記述されているプログラムの入出力に関する情報を第11図のソースプログラム解析情報ファイル33へ出力する。

第12図のプログラム例を解析して得られたソースプログラム解析情報ファイル33内の情報部を第13図に示す。次に第11図ソースプログラム解析情報統合部34によりJCL解析情報ファイル3とソースプログラム解析情報ファイル33を入力し、各々のファイル中のプログラム名称とファイル名称を基に整合性を行ない、JCL解析

情報ファイル中の入出力区分がないファイルに対しソースプログラム解析情報ファイル33より得られた入出力区分の情報を付加する。

第14図の場合JCL解析情報ファイルのプログラム名称“PROGRAM1”、ファイル名称“FILE2”の入出力区分が不定であるが、ソースプログラム解析情報ファイルより同一プログラム名称と同一ファイル名称を持つ行39を検索し、JCL解析情報ファイルに当該行の入出力区分の情報を埋めている。

以下第1図の実施例と同様の手段によりデータフロー仕様書を作成することができる。

上記実施例の他にジョブ内のプログラム実行順序及びプログラムの入出力ファイルを図式的に記述したシステムフロー仕様書からデータフロー仕様書を作成、データベース記述言語によって記述されるデータベース定義情報よりデータベース内の構成員の関連を表すE. R(Entity,Relation)仕様書を作成、オンラインプログラムのソースプログラムよりオンラインプログラムの画面名称と

画面遷移状態を表す画面遷移仕様書を作成する事が本方式で実現できる。

#### 【発明の効果】

本発明によれば、作成されたソフトウェア生産物(プログラム、JCL、下位仕様書等)から上位の仕様書を自動生成できるため、上位仕様書の内容と下位のソフトウェア生産物との整合性が保証されるのでソフトウェア保守作業の効率化が図れる。

#### 4. 図面の簡単な説明

第1図は本発明の一実施例の機能ブロック図、第2図は本発明のハードウェア環境を示す図、第3図は本発明の一実施例の入力となるJCLを示す図、第4図はJCL解析結果を示す図、第5図はJCL解析結果をファイル/表名名称に着目してジョブによる参照/更新関係にまとめ直した図、第6図は第5図のファイル/表名名称のジョブによる参照/更新関係をネットワーク状に展開し図式的に表した図、第7図は第6図のネットワーク情報の各ノードを形式変換して得られたデータフ

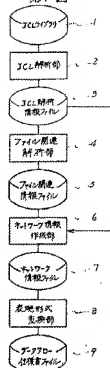
ロードダイアグラムを示す図。第8図は本発明の第2図の実施例の機能ブロック図、第9図は第2図の実施例の入力となるJCLを示す図、第10図は第9図のJCL例の解析結果を示す図、第11図は本発明の第3の実施例の機能ブロック図、第12図は入力となるソースプログラムの例を示す図、第13図はソースプログラムの解析結果を示す図、第14図はプログラム解析結果により不足情報を持った例を示す図である。

1…JCLライブラリ、2…JCL解析部、3…JCL解析情報ファイル、4…ファイル関連解析部、5…ファイル関連ファイル、6…ネットワーク情報作成部、7…ネットワーク情報ファイル、8…表現形式変換部、9…データフロー仕置きファイル。

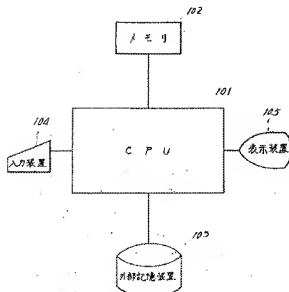
代理人弁護士 小川 肇



第1図



第2図



第3図

(a)

```
< JCL1 : ジョブ名 "JOB1" 実行用 JCL >
10 ~ JOB JOB1
11 ~ ASN EQUIP=DISK, FILE=FILE1, I/O=INPUT
12 ~ ASN EQUIP=DISK, FILE=FILE2, I/O=OUTPUT
13 ~ ASN EQUIP=PRINTER, ... OUTCLASS=A
14 ~ EXEC PROGRAM1
15 ~ END
```

(b)

```
< JCL2 : ジョブ名 "JOB2" 実行用 JCL >
JOB JOB2
ASN EQUIP=DISK, FILE=FILE2, I/O=INPUT
ASN EQUIP=DISK, FILE=FILE3, I/O=OUTPUT
EXEC PROGRAM2
END
```

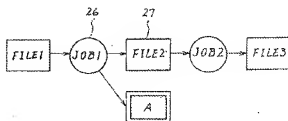
第4図

ジョブ名称	入出力分	ファイル名称	帳票名称
JOB1	入力	FILE1	A
	出力	FILE2	
	出力	FILE3	
JOB2	入力	FILE2	
	出力	FILE3	

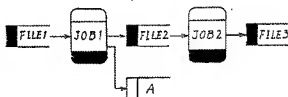
第5図

	ファイル/帳票名称	更新ジョブ名称	参照ジョブ名称
ファイル	FILE1		JOB1 ~22
	FILE2	JOB1	JOB2 ~23
	FILE3	JOB2	~24
帳票	A	JOB1	~25

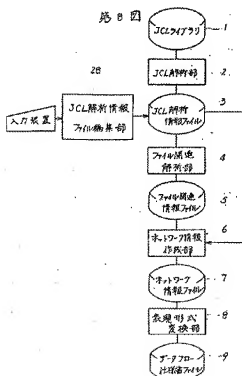
第6図



第7図



第8図



第9図

(A)

&lt; JCL1 : ジョブ名 "JOB1" 実行用JCL &gt;

```

JOB JOB1
ASN EQUIP=DISK.FILE=FILE1.I/O=INPUT
ASN EQUIP=DISK.FILE=FILE3.~29
ASN EQUIP=PRINTER...OUTCLASS=A
EXEC PROGRAM1
END

```

(B)

&lt; JCL2 : ジョブ名 "JOB2" 実行用JCL &gt;

```

JOB JOB2
ASN EQUIP=DISK.FILE=FILE2.I/O=INPUT
ASN EQUIP=DISK.FILE=FILE3.I/O=OUTPUT
EXEC PROGRAM2
END

```



第10図

ジョブ名	プログラム名	入出力	ファイル名	帳票名
JOB1	PROGRAM1	入力	FILE1	A
		出力	FILE2	
JOB2	PROGRAM2	入力	FILE2	
		出力	FILE3	

30

第12図

&lt; PROGRAM1サブプログラム &gt;

```

main() {
35- fopen("FILE1", input);
36- fopen("FILE2", output);
37- fopen("A", output);

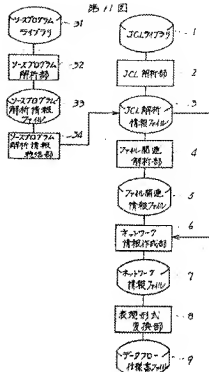
```

データ処理部

第13図

プログラム名	入出力	ファイル名	帳票名
PROGRAM1	入力	FILE1	A
	出力	FILE2	
	出力		

第11図



第14図

